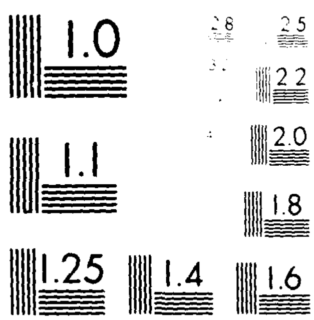
AD-A112 858 GEORGIA INST OF TECH ATLANTA PRODUCTION AND DISTRIBU--ETC F/G 12/1
A BRANCH AND BOUND BASED HEURISTIC FOR SOLVING THE QUADRATIC AS--ETC(U)
OCT 81 M S BAZARAA, O KIRCA N00014-80-K-0709
UNCLASSIFIED PDRC-81-13 NL

1 OF 1
AD A
112858

END
DATE
FILMED
04-82
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

(12)

AD A112858

PRODUCTION and DISTRIBUTION RESEARCH CENTER

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

82 03 01 048

SCHOOL OF INDUSTRIAL
AND SYSTEMS
ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
A UNIT OF THE UNIVERSITY SYSTEM
OF GEORGIA
ATLANTA, GEORGIA 30332

PDRC Report Series
No. 81-13
October, 1981

12

A Branch and Bound Based Heuristic for Solving the Quadratic Assignment Problem

M. S. Bazaraa and O. Kirca

PDRC-81-13

Contract No. N00014-80-k-0709

A Branch and Bound Based Heuristic for Solving
the Quadratic Assignment Problem

M. S. Bazaraa and O. Kirca

Abstract

In this paper a branch and bound algorithm is proposed for solving the quadratic assignment problem. Using symmetric properties of the problem, the algorithm eliminates "mirror image" branches, thus reducing the search effort. Several routines that transform the procedure into an efficient heuristic are also implemented. These include certain 2-way and 4-way exchanges, selective branching rules, and the use of variable upper bounding techniques for enhancing the speed of fathoming.

The computational results are quite encouraging. As an exact scheme, the algorithm solved the 12 facility problem of Nugent et al and the 19 facility problem of Elshafei. More importantly, as a heuristic, the procedure produced the best known solutions for all well-known problems in the literature, and produced improved solutions in several cases.

This research is supported under NSF grant ENG-79-08375 and ONR grant N00014-80-k-0709.

1. INTRODUCTION

The quadratic assignment problem, as given by Koopmans and Beckmann (1957), can be formulated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{\ell=1}^{m} f_{ik} d_{j\ell} x_{ij} x_{k\ell} \tag{1}$$

$$\text{Subject to:} \quad \sum_{i=1}^{m} x_{ij} = 1 \qquad j=1,\ldots,m \tag{2}$$

$$\sum_{j=1}^{m} x_{ij} = 1 \qquad i=1,\ldots,m \tag{3}$$

$$x_{ij} = 0 \text{ or } 1 \qquad i,j=1,\ldots,m \tag{4}$$

The above problem can be interpreted as follows. There are m indivisible objects to be assigned to m indivisible locations, where f_{ik} is the flow or interaction between objects i and k and $d_{j\ell}$ is the distance between locations j and ℓ. The objective is to assign the objects to the locations such that the sum of pairwise interactions among objects weighed by the distance between their respective locations is minimized. Without loss of generality it is assumed that the interaction and distance matrices are symmetric.

There exists two approaches for solving the quadratic assignment problem exactly. The first approach utilizes the concept of branch and bound or implicit enumeration, as in the works of Gilmore (1962), Lawler (1963), Graves and Whinston (1970), Bazaraa and Elshafei (1979), Burkard and Stratmann (1978), Roucairol (1978), Pierce and Crowston (1971), Land (1963), and Gavett and

Plyter (1966). Secondly, through an appropriate transformation, the problem can be reformulated as a linear mixed-integer program which is solved by cutting planes or by a suitable mixed-integer programming package. The algorithms of Bazaraa and Sherali (1980), Kaufman and Broeckx (1978), and Love and Wong (1976) fall into this class.

Due to the complexity of the quadratic assignment problem, in general, none of the above methods can solve problems with dimension $m > 15$ effectively. Thus for larger problems, a considerable amount of effort has been given to the development of inexact methods that obtain good quality solutions with a reasonable computational effort. A comprehensive survey of inexact methods can be found in the works of Sherali (1979) and Burkard and Stratmann (1978). A summary of inexact methods for solving the quadratic assignment problem is given below.

a) Construction Methods

Starting with a partial solution or the null assignment, a complete assignment is reached iteratively by locating one or more objects at each iteration.

b) Improvement Methods

Starting with a complete assignment of objects, an improvement over the incumbent objective function value is sought by interchanging the locations of several objects. The procedure is terminated when no further improvements are possible.

c) Hybrid Methods

Methods in this class combine several features of exact and inexact procedures.

According to the computational experience reported in the literature, it seems that hybrid methods are emerging as the most successful approach for solving large quadratic assignment problems. Examples of such procedures are the methods of Bazaraa and Sherali (1980) and Burkard and Stratmann (1978). Both methods use an exact solution scheme in conjunction with some improvement procedures. Bazaraa and Sherali implement Benders' partitioning method to a mixed-integer formulation of the problem and apply several improvement procedures to the solutions found throughout the course of partitioning. The method of Burkard and Stratmann alternates between a branch and bound (Perturbation) routine and an exchange routine (Verbes) until no better solutions can be obtained.

In this paper, a branch and bound algorithm for solving the quadratic assignment problem is proposed. The main feature of the procedure is the elimination of "mirror image" branches in the search tree. The branch and bound procedure is modified in order to accelerate the computations resulting in an efficient heuristic procedure with the following characteristics:

1. Several improvement routines are used in conjunction with the branch and bound scheme. The extent of using these improvement routines is a function of the branch and bound tree level.

2. Several heuristics are utilized to eliminate the search effort at branches which are likely not to lead to objective value improvements. Furthermore, variable upper bounds are used to reduce the number of solutions examined.

The computational results are quite encouraging. As an exact procedure, the algorithm solved the 12 facility problem of Nugent et al (1968) and the

19 facility problem of Elshafei (1977). More importantly, as an inexact procedure, the modified branch and bound algorithm produced the best known or improved solutions for all well-known problems in the literature of the quadratic assignment problem.

2. AN EXACT BRANCH AND BOUND PROCEDURE

Branch and bound procedures for the quadratic assignment problem can be classified into single assignment algorithms, pair assignment algorithms, and pair exclusion algorithms. At each stage of a single assignment algorithm, one unassigned object is assigned to an unoccupied location. The procedures of Gilmore (1962), Lawler (1963), Graves and Whinston (1970), Burkard and Stratmann (1978), and Bazaraa and Elshafei (1979) are some examples of single assignment algorithms. The pair assignment algorithms proceed by simultaneously locating two unassigned objects to two vacant locations. The procedures proposed by Land (1963) and Gavett and Plyter (1966) are of this type. Pierce and Crowston (1971) proposed a pair exclusion procedure where the algorithm proceeds on the basis of a stage-by-stage exclusion of assignments from a solution to the problem.

The proposed procedure is of the single assignment type where the following general approach is pursued. Let:

$X = \{x\text{: } x \text{ satisfies } (2), (3), (4)\}$

I = set of assigned objects

$\bar{I}$ = complement of I, that is, set of unassigned objects

$\sigma(i)$ = location to which object $i \varepsilon I$ is assigned

$J = \{\sigma(i)\text{: } i \varepsilon I\}$

$\bar{J}$ = complement of J, that is, set of vacant locations

$P = (I,J) = \{(i,\sigma(i))\text{: } i \varepsilon I\}$

$X_P = \{x\text{: } x \varepsilon X \text{ and } x_{ij} = 1 \text{ for all } (i,j) \varepsilon P\}$

μ^* = upper bound on the value of the objective function

π^* = assignment vector of the objects corresponding to the upper bound μ^*, $\pi^* \varepsilon X$.

At each stage of the procedure, we select a partial assignment of objects I and locations J that form the partial assignment set P. The set X_P is then partitioned into $x_{P_1}, x_{P_2}, \cdots, x_{P_n}$ such that:

$$X_{P_k} \cap X_{P_\ell} = \emptyset \quad \text{if } k \neq \ell; \quad k,\ell = 1,\ldots,n$$

$$\bigcup_{k=1}^{n} X_{P_k} = X_P$$

For a selected partial assignment P_k, a lower bound Z_{P_k} is computed. If $Z_{P_k} \geq \mu^*$ then P_k is fathomed, that is, discarded from further considerations. Otherwise, a complete assignment π_{P_k} is sought and its corresponding objective value μ_{P_k} is calculated. If $\mu_{P_k} < \mu^*$ then μ^* and π^* are updated to μ_{P_k} and π_{P_k}, respectively. The above procedure is repeated until no partial assignment P whose lower bound is less than μ^* can be found.

The process of partitioning the X_P into $X_{P_1}, X_{P_2}, \cdots, X_{P_n}$ is referred to as <u>branching</u> from the <u>node</u> representing the partial assignment P. The number of objects in the partial solution is called the <u>level</u> of the tree. The <u>active nodes</u> or <u>active branches</u> is the set of all partial solutions that have not been fathomed or selected for further branching. A branch and bound scheme for solving the quadratic assignment problem can be fully described by specifying rules for:

1) Computing a lower bound Z_P on the objective value of all completions of a partial solution P.

2) Choosing an active node (partial solution) for branching.

3) Branching from a selected partial assignment.

There exist several lower bounding procedures such as those of Gilmore (1962), Lawler (1963), Graves and Whinston (1970), Roucairol (1978), Edwards (1980), Christofides et al (1980) and Frieze and Yadegar (1981). Considering the strength of the bounds and the computational effort, the procedure of Gilmore-Lawler seems to be the most effective. This procedure is adopted here and is described briefly as follows.

Given a partial assignment $P = (I,J)$, the lower bound Z_p is obtained by solving the following linear assignment problem LAP:

$$Z_P = \underset{x \in X_P}{\text{Minimum}} \sum_{i \in \bar{I}} \sum_{j \in \bar{J}} w_{ij} x_{ij} + \nu_P \tag{6}$$

where:

$$w_{ij} = 2 \sum_{k \in I} f_{ik} d_{j\sigma(k)} + < \bar{f}(i), \bar{d}(j) >$$

$\bar{f}(i)$ = vector of interactions of object i with other unassigned objects in $\bar{I}$, where the elements of the vector are sorted in an ascending order.

$\bar{d}(j)$ = vector of distances from location j to other unoccupied locations in $\bar{J}$, where distances are sorted on a descending order.

$$\nu_P = \sum_{i \in I} \sum_{k \in I} f_{ik} d_{\sigma(i)\sigma(k)}$$

$<\cdot, \cdot>$: stands for the inner product of two vectors.

In the above linear assignment problem, w_{ij} is a lower bound on the assignment of object $i \in \bar{I}$ to location $j \in \bar{J}$. The fixed cost ν_p is the value

accruing from the current assignment of objects in I to locations in J. Let the optimal assignment of objects in problem LAP be a(i) for $i \in \bar{I}$. Then:

1) Z_P is a lower bound on the objective value of all completions of the partial assignment P.

2) The quadratic cost μ_P of the solution π_P given below can be used to update μ^* provided that $\mu_P < \mu^*$.

$$\pi_P(i) = \begin{cases} \sigma(i) & \text{if } i \in I \\ a(i) & i \in \bar{I} \end{cases}$$

3. At optimality of problem LAP, a set of Lagrangian multipliers u_i for $i \in \bar{I}$ and v_j for $j \in \bar{J}$ with the following properties, is available:

$$w'_{ij} = w_{ij} - u_i - v_j \geq 0 \quad i \in \bar{I},\ j \in \bar{J}$$

$$w'_{ij} = 0 \qquad \text{if} \quad x_{ij} = 1$$

The reduced costs w'_{ij} for $i \in \bar{I}$, $j \in \bar{J}$ can be utilized to compute lower bounds for all branches emanating from the node associated with the partial solution P = (I,J) without the need for solving new linear assignment problems. This procedure is called the alternative cost method and has been applied by Little et al (1963) for the travelling salesman problem and later used by Pierce and Crowston (1971) for the quadratic assignment problem.

To demonstrate the use of the alternative cost principle, consider the partial assignment P = (I,J). Let b(j) be the object assigned to location $j \in \bar{J}$ in the solution to problem LAP. Now consider the branch with the partial assignment

$$P' = P \cup \{(r,s)\} \quad \text{for} \quad r \varepsilon \bar{I} \text{ and } s \varepsilon \bar{J}$$

A lower bound $\bar{Z}_{P'}$ on the objective values of all completions of P' is readily available as:

$$\bar{Z}_{P'} = Z_P + \gamma_{rs}$$

where:

$$\gamma_{rs} = \begin{cases} 0 \text{ if } \quad s = a(r) \\ \\ \text{minimum } \{w'_{rs} + w'_{b(s)\ a(r)},\ w'_{rs} + \alpha_{b(s)} + \beta_{a(r)}\} \end{cases}$$

$$\alpha_i = \underset{\substack{\ell \varepsilon \bar{J} \\ \ell \neq a(i)}}{\text{minimum}}\ w'_{i\ell}$$

$$\beta_j = \underset{\substack{\ell \varepsilon \bar{I} \\ \ell \neq b(\bar{j})}}{\text{minimum}}\ w'_{\ell j}$$

2.1 Selection of the Branching Node

At each stage of the branch and bound procedure, a partial assignment has to be selected among all active branches. The following two strategies are typically used:

1) Depth First

Choose the active branch with the least lower bound among the most recently created active branches.

2) Breadth First

Choose the active branch with the least lower bound among all active branches in the current decision tree.

The attainment of good quality solutions early on is of great importance for the quadratic assignment problem, especially if the algorithm is eventually used as a heuristic. Implementing the depth or breadth strategies alone is not satisfactory. The correlation between lower bounds and quality of partial assignments at low levels of the branch and bound tree is not strong. Thus it is highly likely that a depth strategy may select poor quality branches to pursue initially so that good quality solutions are obtained only after a large number of nodes is evaluated. On the other hand, high levels of the branch and bound tree are not reached early on if the breadth strategy is used. Since good quality solutions are usually obtained only at high levels of the tree, the process of obtaining such solutions is also delayed. For this reason, the proposed algorithm combines the two branching strategies. Particularly, a breadth strategy is used as long as the tree level L has not reached L_1 for the first time, where L_1 is a suitable trigger parameter. The depth stragegy is implemented if $L > L_1$. With this combined strategy many candidate good quality partial assignments are formed at low tree levels. Starting with one of these solutions, the depth strategy quickly finds good quality complete solutions. If L_1 is set equal to 0, then the proposed procedure reduces to depth first, and if L_1 is set equal to m, it reduces to breadth first.

The choice of the trigger parameter L_1 is highly dependent on the dimension of the problem. A large value of L_1 increases the computer storage requirements as well as delay the attainment of good quality solutions. According to our computational experience and depending on the problem size, values of L_1 from 3 to 5 are found to be satisfactory.

2.2. Branching from an Active Node

In the proposed branch and bound procedure, the single assignment rule is used for branching from a selected active node. Particularly, an object $r \varepsilon \bar{I}$ is selected and $|\bar{J}|$ branches each corresponding to $x_{rs} = 1$ for $s \varepsilon \bar{J}$ are formed. As described previously, by using the alternative costs, some of these branches may be fathomed immediately. Some alternative procedures for selecting the particular object r for branching are given below.

2.2.1. Select object r using alternative costs

Alternative costs can be used in the process of selecting the branching object. By the use of alternative costs, it is possible to estimate the rate of increase of lower bounds associated with each object $i \varepsilon \bar{I}$. An object r in $\bar{I}$ is selected according to one of the following two rules:

1) Maximum total alternative cost rule

Choose $r \varepsilon \bar{I}$ satisfying

$$\sum_{j \varepsilon \bar{J}} \gamma_{rj} = \underset{i \varepsilon \bar{I}}{\text{maximum}} \sum_{j \varepsilon \bar{J}} \gamma_{ij}$$

Here, object r that results in the maximum sum of all lower bounds at the next tree level is selected.

2) Minimum number of branches rule

Choose $r \varepsilon \bar{I}$ satisfying

$$\sum_{j \varepsilon \bar{J}} \delta_{rs} = \underset{i \varepsilon \bar{I}}{\text{minimum}} \sum_{j \varepsilon \bar{J}} \delta_{rj}$$

where,

$$\delta_{ij} = \begin{cases} 0 \text{ if } Z_p + \gamma_{ij} \geq \mu^* \\ 1 \text{ otherwise} \end{cases}$$

Here, object r that results in the minimum number of branches at the next tree level is selected.

The proposed procedure uses a combination of the above rules. First a branching object is attempted using rule (2). If $\sum_{j\varepsilon J} \delta_{rj} < |\bar{J}|$, then object r is selected. Otherwise object r is selected using rule (1).

2.2.2. Select object r using a predetermined order

Here objects are ranked with respect to a certain criterion and object $r\varepsilon\bar{I}$ with the highest rank is selected. The following are some possible criteria for ranking the objects:

1) Maximum total interaction with all objects.
2) Maximum total interaction with unassigned objects.
3) Maximum total interaction with assigned objects.

Our computational experience suggests that selecting objects using alternative costs is superior.

2.3. Elimination of Mirror Image Branches

Two assignments π_1 and π_2 are mirror images in a quadratic assignment problem if the following hold:

1) $\pi_1 \neq \pi_2$
2) $d_{\pi_1(i)\pi_1(k)} = d_{\pi_2(i)\pi_2(k)}$ for all $i,k=1,\ldots,m; i\neq k$

In a quadratic assignment problem where the distance matrix corresponds to a rectangular layout, it is possible to identify several mirror images of a given assignment of objects. A mirror image of an assignment can be obtained by rotating the objects column-wise or row-wise such that all pairwise

distances among objects remain unchanged. Hence both assignments have the same quadratic objective function value. An example of obtaining mirror image assignments is given for a 2x4 layout in Figure 1.

The above property holds for certain partial assignments also. Two partial assignments $P_1 = (I_1, J_1)$ and $P_2 = (I_2, J_2)$ are mirror images of each other if the following hold:

1) $I_1 = I_2 = I$

2) $d_{\sigma_1(i)\sigma_1(k)} = d_{\sigma_2(i)\sigma_2(k)}$ for all $i,k \epsilon I$ $i \neq k$

3) For every $j \epsilon \bar{J}$, there exists a location $\ell \epsilon \bar{J}_2$ where:

(i) $d_{j\sigma_1(k)} = d_{\ell\sigma_2(k)}$ for all $k \epsilon I$

(ii) $\bar{d}_1(j) = \bar{d}_2(\ell)$

Condition (1) assures that both partial solutions involve the same set of objects. The second condition asserts that all pairwise distances among assigned objects are equal in both partial assignments. The last condition states that for an unoccupied location $j \epsilon \bar{J}$, there exists another unoccupied location $\ell \epsilon \bar{J}_2$ such that the respective distances to locations of assigned objects and to vacant locations are equal. Obviously, if conditions (1)-(3) are satisfied then the respective lower bounds Z_{P_1} and Z_{P_2} will be equal also. Furthermore, all higher level branches emerging from P_1 and P_2 will also

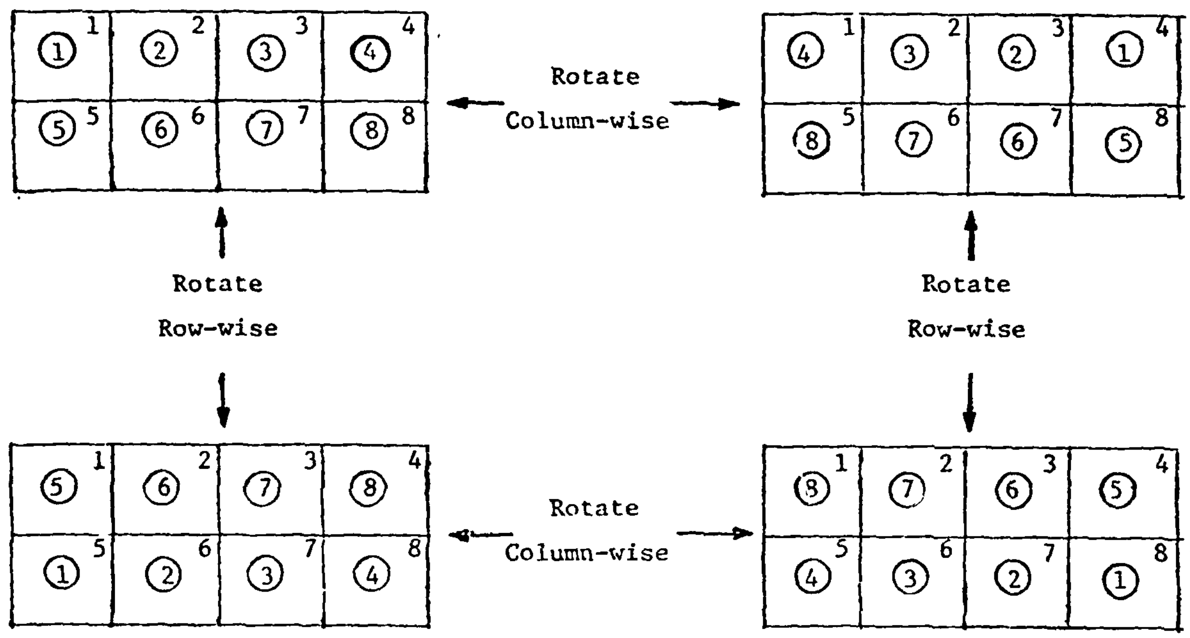


Figure 1. Mirror images of an assignment on a 2x4 layout

be mirror images of each other. Thus whenever two mirror image partial assignments are observed, one of the associated branches can be fathomed immediately.

A substantial reduction in the number of partial assignments can be achieved using the mirror image property. For example, in the 12 facility problem of Nugent et al (1968), at level 1, instead of forming 12 branches it is sufficient to create only 4 branches, resulting in a reduction of 66% in the computational effort.

The above branch and bound procedure, termed EXBB, is coded in Fortran IV. EXBB is applied to some standard quadratic assignment problems in the literature and the computational results are summarized in Table 1. In Table 2 these results are compared with other branch and bound procedures. A considerable amount of reduction in the total search effort (number of nodes and number of LAP's) is achieved with EXBB for the test problems. Also note that for QAP7 where the locations permit no mirror image assignments, the difference between the performance of EXBB and the other two methods is not clear.

Table 1. Computational Results of EXBB

Problem		Dimension m	Number of Nodes (1)	Number of LAP's solved	Optimal Objective Value	Time Cpu Sec (2)
Nugent, Vollmann and Ruml Problems [20]	QAP5	5	3	3	25	.023
	QAP6	6	5	5	43	.048
	QAP7	7	58	36	74	.372
	QAP8	8	39	28	107	.466
	QAP12	12	3385	2201	289	78.220
	QAP15	15	16001	12269	575	500.00 (3)
Elshafei's Problem (7)	QAP19	19	767	715	8606274	109.027

(1) Using branching object selection strategy alternative costs as described in Section 2.2.1

(2) On a CDC Cyber 70 Model 74-28/CDC 6400

(3) Terminated at that time without verifying optimality

Table 2. Comparison of EXBB with some other Branch and Bound Procedures

Problem	EXBB		Burkard and Stratmann (1978)		Bazaraa and Elshafei (1979)	
	Number of Nodes	Number of LAP's Solved	Number of Nodes	Number of LAP's Solved	Number of Nodes	Number of LAP's Solved
QAP5	3	3	Not Available	8	20	14
QAP6	5	5	Not Available	25	67	36
QAP7	58	36	Not Available	28	73	40
QAP8	39	28	Not Available	189	235	141
QAP12	3385	2201	Not Available	15575	37531	26368

3. AN INEXACT METHOD BASED ON BRANCH AND BOUND

When the exact branch and bound procedure is applied to the test problems in the last section, it is observed that the optimum solutions are reached early on in the search procedure. As shown in Table 3, the remaining effort is spent to prove optimality of the solution.

Table 3. Effort Spent to Prove Optimality

Problem	Total number of nodes	Node # at which optimal solution is found	% effort spent in proving optimality
QAP7	58	6	89
QAP8	39	13	66
QAP12	3385	535	84
QAP19	767	78	89

In this section, several improvement routines and methods of eliminating certain branches which are likely not to produce good quality solutions are discussed. With these modifications, the branch and bound scheme is transformed into a heuristic that produces good quality solutions within a reasonable computational effort. The major revisions to EXBB are:

1) In order to improve the quality of upper bounds, exchange routines are applied to the LAP solutions at certain branches in the search tree.

2) Since it is not possible to exhaust the search tree for large problems, several heuristics are developed for discontinuing the search at

branches where improvements are not likely even if the lower bounds indicate that fathoming is not yet achieved.

3.1. Application of the Exchange Routine

An attempt to improve the quality of the upper bound is made by applying an exchange routine to some of the LAP solutions obtained in the lower bounding process. The application of the improvement routine to all LAP solutions is not advisable. Especially at low levels of the tree, the quality of the LAP solutions is not good, so that even with the exchange routine it is usually not possible to update the upper bound.

Two levels n_1 and n_2 where $n_1 \leq n_2$ are selected. These parameters are used to trigger the exchange algorithm as follows:

1) At levels $L \leq n_1$ the exchange routine is applied to all LAP solutions. Even though at this stage it is not likely to obtain good quality solutions, the exchange routine is implemented in order to improve the solutions for use in conjunction with the variable upper bounds (Section 3.5).

2) At levels $n_1 < L < n_2$, the routine is applied only at the branch that has the least objective function value μ_p among all the branches at that level.

3) At levels $L \geq n_2$, the routine is applied at all branches in the hope of improving the quality of solutions at hand.

A suitable choice of the parameter n_1 is 1 and a good value of n_2 is around m/2.

3.2. The Exchange Routine

Most of the improvement algorithms available in the literature utilize 2-way exchanges with different implementation strategies. Some routines also use the higher order 3-way and 4-way exchanges. Los (1978) and Burkard and Stratmann (1978) discussed some procedures which employ higher order exchanges and conclude that the extra effort spent is not worthwhile.

In general, 4-way exchanges are computationally very expensive. Given an assignment of four objects (i,k,p,q), there are 23 different additional permutations. Mirchandani and Obata (1979) showed that out of these 23 permutations, 6 can be obtained by 2-way exchanges and 8 can be obtained by 3-way exchanges. Thus, only 9 permutations require 4-way exchanges. Three of these remaining 9 permutations can be easily computed by making use of 2-way exchanges.

The simultaneous exchange of locations of two pairs of objects is called a 2x2-way exchange. Specifically, consider two pairs of objects (i,k) and (p,q). Let $\Delta(i,k)$ and $\Delta(p,q)$ be the change in the objective function value for the 2-way exchanges of these two pairs of objects, respectively. Then the net change $\Delta(i,k,p,q)$ in the objective function value resulting from exchanging the locations of objects i and k and those of p and q is given below:

$$\begin{aligned}\Delta(i,k,p,q) &= \Delta(i,k) + \Delta(p,q)\\ &\quad + (f_{ip} + f_{kq} - f_{kp} - f_{iq})(d_{a(k)a(p)} + d_{a(i)a(q)} - d_{a(i)a(p)} - d_{a(k)a(q)})\end{aligned}$$

where a(i) is the location of object i in the current assignment.

The exchange routine implemented in this study evaluates 2-way and 2x2-way exchanges in the following way:

1) First improvement rule is adopted. That is, the first exchange that yields an improvement is implemented.

2) An exchange of two objects is considered only if the distance between their respective locations does not exceed a certain parameter λ.

3) The objects are ranked according to total interactions, and exchanges are performed starting with objects at the top of the list.

Using the above rules, 2-way exchanges are first performed until no improvements can be obtained. Then 2x2-way exchanges are evaluated. If any 2x2-way exchange results in a smaller objective value, the routine is reinitiated using 2-way exchanges starting from the improved solution. This procedure is terminated when no improvements are possible using the 2x2-way exchange routine.

3.3. Selective Location Rule

In optimal or good quality solutions it is generally expected that objects with large total interaction are assigned to "median" locations while objects with small total interaction are assigned to "off-median" or corner locations. While branching from a partial assignment, new branches are created by assigning the selected object to each of the vacant locations. The assignment of a high interaction object to a corner location is likely not to lead to a good quality solution, even if the lower bound does not exceed the incumbent objective value. In general, a substantial computational effort may be expended in pursuing such "bad" branches. By the selective location rule, high interaction objects are assigned only to "cental" locations and low interaction

objects are assigned to "off-median" locations. Obviously, there is no guarantee that all optimal or good quality solutions must satisfy these additional restrictions, but it is hoped that the exchange routine would help overcome this difficulty.

The objects and locations are ranked according to non-increasing total interactions and non-decreasing total distances, respectively. A parameter t is chosen and f… each object i, the set of permissible locations T_i is determined as follows. Let i^* be the rank of the object i. Then:

$$T_i = \{J: i^* - t \leq j^* \leq i^* + t\}$$

where j^* is the rank of location j. A good choice of t is in the range from $m/3$ to $m/2$.

3.4. Group Assignment of Objects

In an optimal solution we would generally expect that the set of objects with large pairwise interactions to be located close to each other. This observation is incorporated in the branch and bound procedure as follows. Choose a parameter ε denoting the threshold percentage of total interactions. Suppose that at branch P object r is assigned to location s, and that the solution of LAP yields the complete assignment π_P. Then for all $i \varepsilon \bar{I}$ satisfying:

$$\frac{f_{ri}}{\sum_{k=1}^{m} f_{rk}} \geq \varepsilon$$

object i is assigned to the location $\pi_P(i)$ for all completions of the current

partial assignment P. That is, P is updated as follows:

$$P \leftarrow P \cup \{(i,\pi_p(i)\}$$

By this procedure it is hoped to speed the branch and bound scheme and at the same time generate good partial assignments. A suitable choice of ε is around .25.

3.5. Variable Upper Bounds

In order to reduce the search effort, variable upper bounds are used. Bazaraa and Elshafei (1977) discussed fictitious upper bounding procedures for tree search algorithms and applied them to the quadratic assignment problem in [2]. Also, Burkard and Stratmann (1978) implemented a variable upper bounding scheme to an inexact branch and bound procedure for the same problem. The concept of fictitious or variable upper bounding can be explained as follows.

A branch is fathomed if its lower bound is greater than or equal to the incumbent upper bound. But in quadratic assignment problems, the upper bound usually exceeds the lower bound except for large tree levels. Thus a substantial amount of effort is typically spent is pursuing bad quality solutions before fathoming can be achieved. In order to speed fathoming, a fictitious upper bound $V(L) \leq \mu^*$ is set for each level L of the search tree. The branches at level L that have a larger lower bound than V(L) are fathomed. These fictitious upper bounds are called variable upper bounds since their values depend on the tree level L, where $V(L) \leq V(L + 1)$ for $L=1,\cdots,m-1$. The variable upper bounds must be such that good quality branches are not fathomed. Only the branches which are likely not to produce improved solutions are fathomed.

In order to determine the form of the variable upper bounds, the gap between the lower bounds and the best known solutions for some test problems are examined. As Burkard and Stratmann (1978) suggested, the gap decreases quadratically with respect to the level of the tree. A proper function for the variable upper bounds V(L) is of the form:

$$V(L) = \begin{cases} Z_0 + g_1 L + g_2 L^2 & L=1,\ldots,\theta \\ \mu^* & L=0 \text{ or } L>\theta \end{cases}$$

where θ is a suitable parameter corresponding to the tree level at which lower and upper bounds are usually equal. The values of coefficients g_1 and g_2 are determined by the following boundary conditions:

$$V(\theta) = \mu^*$$

$$2V(\theta/2) - Z_0 = \mu^*$$

The computational experience with variable upper bounds show that relatively good quality solutions are obtained early on and further efforts either do not improve the current upper bound or produce little improvement. Since for a fixed L, an increased θ results in decreasing V(L) and hence speeding the fathoming process, the parameter θ is incremented by one after evaluating certain number of nodes. For Gilmore-Lawler bounds, a good starting value for θ is around $\frac{2}{3}m$. Furthermore, the branch and bound procedure is terminated if no improvements over the current upper bound are obtained within a specified amount of time.

The exact branch and bound procedure EXBB of Section 2 is modified using the above heuristic strategies, resulting in the code INBB. A variant of this inexact procedure that differs only in the computation of lower bounds is also developed. INRO uses Roucairol's reduction procedure [22]. The basic idea of Roucairol's procedure is to reduce the interaction and distance matrices so that the quadratic assignment problem is written as:

$$\text{Minimize} \quad \sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{k=1}^{m} \sum_{\ell=1}^{m} f'_{ik} d'_{j\ell} x_{ij} x_{k\ell} + \sum_{i=1}^{m} \sum_{k=1}^{m} c_{ij} x_{ij} + \nu_0$$

$$\text{Subject to:} \quad x \varepsilon X$$

where,

f'_{ik} = nonnegative reduced flow from object i to object k

$d'_{j\ell}$ = nonnegative reduced distance between locations j and ℓ

c_{ij} = reduced cost of locating object i to location j

ν_0 = a fixed cost obtained by the reduction process.

Ignoring the reduced quadratic part, the lower bound Z_P for a given partial assignment P is computed by solving the following LAP:

$$Z_P = \underset{x \varepsilon X_P}{\text{Minimum}} \sum_{i \varepsilon \bar{I}} \sum_{j \varepsilon \bar{J}} h_{ij} x_{ij} + \nu_P + \nu_0$$

where;

$$h_{ij} = c_{ij} + \sum_{k \varepsilon I} f'_{ik} d'_{j\sigma(k)}$$

$$\nu_P = \sum_{i \varepsilon I} c_{i\sigma(i)} + \sum_{i \varepsilon I} \sum_{k \varepsilon I} f'_{ik} d'_{\sigma(i)\sigma(k)}$$

Obviously, the above lower bound can be strengthened by computing a lower bound on the reduced quadratic term. But our experience with the test problems suggest that including the reduced quadratic term into the lower bound computations does not necessarily yield stronger lower bounds in comparison with Gilmore-Lawler procedure.[†] Furthermore as far as the total effort is concerned, by including the reduced quadratic relations into the bound computations, the computational advantage of Roucairol's procedure over the Gilmore-Lawler method is lost.

The modified branch and bound schemes INBB and INRO are applied to the problems of Nugent et al (1968). For each problem, the following three strategies for selecting the branching object are used.

1) Maximum total interaction with all objects
2) Maximum total interaction with already assigned objects
3) The alternative cost rule described in Section 2.2.1.

The computational results are given in Table 4. As seen from the table, even with the weaker bounding procedure INRO, good quality solutions are obtained in a relatively small amount of computational time. Furthermore, it is observed that the selection rule of the branching object does not affect the quality of the solutions significantly. An attempt to further improve the quality of the solutions at hand is made using a routine that implements 2, 3, and 4-way exchanges. This routine is similar to the Mixed Exchange Algorithm of Mirchandani and Obata (1979). Either very little improvement or none at all

---

[†] For problems that have a rectangular grid layout and an interaction matrix with at least one zero in each row and column, it can be shown that Roucairol's procedure cannot yield stronger lower bounds even if the reduced quadratic term is included in the bound computations.

Table 4. Summary of the Results for Nugent et al (1968) Problems

Problem	Dimension m	Branching Object Selection Strategy	INRO		INBB		Best known value in literature [3]
			Value	Cpu. Sec. (a)	Value	Cpu. Sec. (a)	
QAP15	15	(1)	576	11.85	575	31.49	
		(2)	576	16.55	575	32.74	575
		(3)	575	10.80	575	30.59	
QAP20	20	(1)	1297	30.75	1285	100.00	
		(2)	1285	29.50	1285	100.00	1285
		(3)	1300	50.00	1285	100.00	
QAP30	30	(1)	3079	176.19	3079	400.83	
		(2)	3083	139.49	3080	346.95	3077
		(3)	3080	200.00	3078	400.84	

(a) On a CDC cyber 70 model 74-28/CDC 6400

is attained as a result of these computationally expensive exchange routines, rendering their use unjustified.

4. AN ITERATIVE APPLICATION OF THE INEXACT BRANCH AND BOUND PROCEDURE

One disadvantage of the proposed inexact methods of Section 3 is that their effectiveness is highly dependent on the quality of the initial partial assignments selected at low tree levels. Since the search tree is not exhausted, these methods commit themselves to the partial solutions selected at low levels. In order to reduce this dependency on initial partial assignments the following iterative branch and bound procedure is developed.

The procedure applies the inexact branch and bound scheme iteratively by alternating between two branching rules. At each iteration, several number of objects are fixed at the locations of the previous iteration and the branch and bound computations are performed by changing the branching rule. The procedure terminates when no improvements are obtained, and is summarized as follows.

Form two ordered sets of objects S and $\bar{S}$ such that:

$$\sum_{k=1}^{m} f_{i_\ell,k} \geq \sum_{k=1}^{m} f_{i_{\ell+1},k} \qquad \text{for all } i_\ell \varepsilon S \qquad \ell=1,\ldots,m$$

and

$$\sum_{k=1}^{m} f_{i_\ell,k} \leq \sum_{k=1}^{m} f_{i_{\ell+1},k} \qquad \text{for all } i_\ell \varepsilon \bar{S} \qquad \ell=1,\ldots,m$$

Let s_q be a parameter corresponding to the branching object selection rule at iteration q, where:

$$s_q = \begin{cases} 1 & \text{select the branching object having maximum total interaction with all objects.} \\ 2 & \text{select the branching object having minimum total interaction with all objects.} \end{cases}$$

<u>Step 0</u>: Set $q=1$, $s_q=2$, $\bar{s}_q=1$, $\mu^*=\infty$, and $\pi^*=0$.

Let $P_0 = (I_0,J_0) = (\emptyset,\emptyset)$ and go to Step 1.

<u>Step 1</u>: Apply INBB using the branching rule s_q, and starting with the partial solution P_0. Terminate the branch and bound procedure whenever the bottom of the search tree is reached for the first time. Let the best solution found be μ_q and π_q and go to Step 2.

<u>Step 2</u>: If $\mu_q > \mu^*$ or $\mu_q = \mu_{q-2}$ stop. Otherwise go to Step 3.

<u>Step 3:</u> Let , $\mu^* = \mu_q$ and $\pi^* = \pi_q$. Determine a number k_q which corresponds to the number of objects to be fixed at this iteration. Set $P_0 = (I_0,J_0)$ as follows:

$$I_0 = \begin{cases} \text{first } k_q \text{ elements of ordered set } S & \text{if } \bar{s}_q=1 \\ \text{first } k_q \text{ elements of ordered set } \tilde{S} & \text{if } \bar{s}_q=2 \end{cases}$$

and

$$J_0 = \{\pi^*(i): \ i\varepsilon I_0\}$$

Let,

$$s_{q+1} \leftarrow \bar{s}_q$$
$$\bar{s}_{q+1} \leftarrow s_q$$
$$q \leftarrow q+1$$

and return to Step 1.

The procedure terminates at an iteration q if the objective value of the assignment found at that iteration is greater than the incumbent μ^* or equal

to that obtained by the previous iteration which has the same branching rule. The number of objects k_q to be fixed at each iteration is around m/3.

The above iterative procedure is applied to Nugent et al (1968) and Steinberg (1961) problems and the results are summarized in Table 5. The procedure produced at the best known solutions for problems QAP20 and QAP34-1. For Problems QAP30 and QAP34-2 the iterative procedure produced solutions which are better than the best known in the literature. These solutions are given in the Appendix.

Table 5. Summary of Iterations for Nugent et al (1968) and Steinberg (1961) Problems with Iterative Branch and Bound Scheme

Problem	Dimension m	Iteration q	Branching Order s_q	Number of fixed objects k_{q-1}	Objective value	Time Cpu. Sec. (a)	Cumulative Computation Time Cpu. Sec. (a)	Best known in literature [3]
QAP20	20	1	2	0	1303	52.62	52.62	1285
		2	1	6	1298	28.32	70.94	
		3	2	6	1298	19.24	90.18	
		4	1	8	1292	20.37	110.55	
		5	2	8	1285	20.08	130.63	
		6	1	10	1285	15.40	156.03	
QAP30	30	1	2	0	3130	130.53	130.53	3077
		2	1	8	3072	61.91	192.44	
		3	2	8	3064	67.11	259.55	
		4	1	12	3064	31.20	290.75	
		5	2	12	3064	29.50	320.25	
QAP34-2 Squared Eucli-dean Distance	34	1	2	0	7999	398.22	398.22	7926
		2	1	13	7926	46.56	444.78	
		3	2	16	7926	62.26	506.71	
QAP34-2 Rectin-linear Distance	34	1	2	0	4802	326.60	326.60	4802
		2	1	13	4800	55.87	382.47	
		3	2	16	4800	83.26	465.73	

(a) On CDC Cyber 70 model 74-28/CDC 6400

APPENDIX

SOLUTIONS FOR QAP30 and QAP34-2

Problem: QAP30

Best found Value: 3064

Assignment:

1 15	2 23	3 11	4 30	5 4	6 14
7 17	8 18	9 8	10 16	11 27	12 20
13 1	14 22	15 7	16 19	17 3	18 29
19 24	20 26	21 10	22 9	23 21	24 2
25 12	26 6	27 25	28 13	29 28	30 5

Problem: QAP34-2

Best found Value: 4800

Assignment:

1 26	2 22	3 27	4 23	5 11	6 6	7 5	8 3	9
10 25	11 21	12 14	13 20	14 12	15 13	16 4	17 8	18 2
19 24	20 32	21 19	22 28	23 1	24 7	25 10	26 18	27 17
28 33	29 34	30 31	31 30	32 29	33 15	34 9	35 16	36

REFERENCES

[1] Bazaraa, M. S. and A. N. Elshafei, "On the Use of Fictitious Bounds in Tree Search Algorithms," Management Science 23, 904-908 (1977).

[2] Bazaraa, M. S. and A. N. Elshafei, "An Exact Branch and Bound Procedure for Quadratic Assignment Problems" Naval Research Logistics Quarterly 26, 109-121 (1979).

[3] Bazaraa, M. S. and H. D. Sherali, "Benders' Partitioning Scheme Applied to a New Formulation of the Quadratic Assignment Problem," Naval Research Logistics Quarterly 27, 29-41 (1980).

[4] Burkard, R. E. and K. H. Stratmann, "Numerical Investigations on the Quadratic Assignment Problem," Naval Research Logistics Quarterly 25, 129-148 (1978).

[5] Christofides, N. A. Mimgozzi, and P. Toth," Contributions to the Quadratic Assignment Problem," European Journal of Operational Research 4, 243-247 (1980).

[6] Edwards, C. S., "A Branch and Bound Algorithm for the Koopmans-Beckman Quadratic Assignment Problem," Mathematical Programming Study 13, 35-52 (1980).

[7] Elshafei, A. N., "Hospital Layout as a Quadratic Assignment Problem," Operational Research Quarterly 28, 167-179 (1977).

[8] Frieze, A. M. and J. Yadegar, "On the Quadratic Assignment Problem," Queen Mary College, March (1981).

[9] Gavett, J. W. and N. V. Plyter, "The Optimal Assignment of Facilities to Locations by Branch and Bound," Operations Research 14, 210-232 (1966).

[10] Gilmore, P. C., "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem," SIAM Journal on Applied Mathematics 10, 305-313 (1962).

[11] Graves, G. W. and A. B. Whinston, "An Algorithm for the Quadratic Assignment Problem," Management Science 17, 453-471 (1970).

[12] Kaufman, L. and F. Broeckx, "An Algorithm for the Quadratic Assignment Problem Using Benders' Decomposition," European Journal of Operational Research 2, 204-211 (1978).

[13] Koopmans, T. C. and M. Beckmann, "Assignment Problems and the Location of Economic Activities," Econometrica 25, 53-76 (1957).

[14] Land, A. M., "A Problem of Assignment with Interrelated Costs," Operational Research Quarterly 14, 185-198 (1963).

[15] Lawler, E. L., "The Quadratic Assignment Problem," Management Science 19, 586-590 (1963).

[16] Little, J. D. C., K. G. Murty, D. W. Sweeney and K. Caroline, "An Algorithm for the Travelling Salesman Problem," Operations Research 14, 972-989 (1963).

[17] Los, M., "Comparison of Several Heuristic Algorithms to Solve Quadratic Assignment Problems of the Koopmans-Beckmann type," Presented at the International Symposium on Locational Decisions, Held at Banff, Alberta, April (1978).

[18] Love, R. F. and J. Y. Wange, "Solving Quadratic Assignment Problems with Rectangular Distances and Integer Programming," Naval Research Logistics Quarterly 23 623-627 (1976).

[19] Mirchandani P. and T. Obata, "Algorithms for a Class of Quadratic Assignment Problems," Presented at the Joint National TIMS/ORSA Meeting, New Orleans, May (1979).

[20] Nugent, C. E., T. E. Vollmann, and J. Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," Operations Research 16, 150-173 (1968).

[21] Pierce, J. F. and W. B. Crowston, "Tree Search Algorithms for Quadratic Assignment Problems," Naval Research Logistics Quarterly 18, 1-36 (1971).

[22] Roucairol, C., "A Reduction Method for Quadratic Assignement Problems," III Symposium Uber Operations Research, Universität Mannheim, October (1978).

[23] Sherali, H. D., "The Quadratic Assignment Problem: Exact and Heursitic Methods," Unpublished Doctoral Dissertation, Georgia Institute of Technology, June (1979).

[24] Steinberg, L., "The Backboard Wiring Problem: A Placement Algorithm," SIAM Review 3, 37-50 (1961).

ATE

LMED

-8